Discrete Mathematics 59 (1986) 99–105
North-Holland

BIJECTIONS BETWEEN FORMULAE AND TREES WHICH ARE COMPATIBLE WITH EQUIVALENCES OF THE TYPE $((f \circ g) \circ h) \sim ((f \circ h) \circ g)$

R.P. NEDERPELT
Department of Mathematics and Computing Science, Eindhoven University of Technology, Eindhoven, The Netherlands

Received 30 May 1985

We describe a bijection which maps trivalent ordered trees, representing certain formulae, onto ordered trees. The mapping is such that an equivalence relation of the type $((f \circ g) \circ h) \sim ((f \circ h) \circ g)$ on the set of formulae, induces the equivalence relation of being equal modulo order on the set of ordered trees.

1. Introduction

Let X be an alphabet and $\circ$ a symbol which does not belong to X. We think of X as a set of operands, and think of $\circ$ as a *binary operation symbol* on X. In this conception we may construct *formulae*, composed of symbols from X, the symbol $\circ$ and brackets. Formally: the set $F(X, \circ)$ of *formulae on X and $\circ$* is recursively defined by:

(i) If $x \in X$, then $x \in F(X, \circ)$;

(ii) If f and $g \in F(X, \circ)$, then $(f \circ g) \in F(X, \circ)$.

Examples of formulae on X and $\circ$ are: x, $(x \circ y)$, $(((x \circ y) \circ x) \circ (z \circ x))$, where x, y, $z \in X$.

In this paper we are interested particularly in an equivalence relation R on $F(x, \circ)$, generated by the following three rules:

(i) If f, g and $h \in F(X, \circ)$, then $((f \circ g) \circ h) R ((f \circ h) \circ g)$;

(ii) (monotony:) If f, g, h and $k \in F(X, \circ)$ such that fRh and gRk, then $(f \circ g) R (h \circ k)$;

(iii) (equivalence:) R is an equivalence relation.

There are two well-known examples of this type of equivalence.

I. Consider algebraic formulae $f, g, h, \ldots$ containing real variables, and take $\circ$ to represent exponentiation. Then $((f \circ g) \circ h)$ and $((f \circ h) \circ g)$ yield the same outcome after calculation for specific values of the variables. Hence, one may regard such formulae as equivalent.

II. Consider logical formulae $f, g, h, \ldots$ and take $\circ$ to represent implication. Then the formulae $(h \circ (g \circ f))$ and $(g \circ (h \circ f))$ are logically equivalent. One may

0012-365X/86/\$3.50

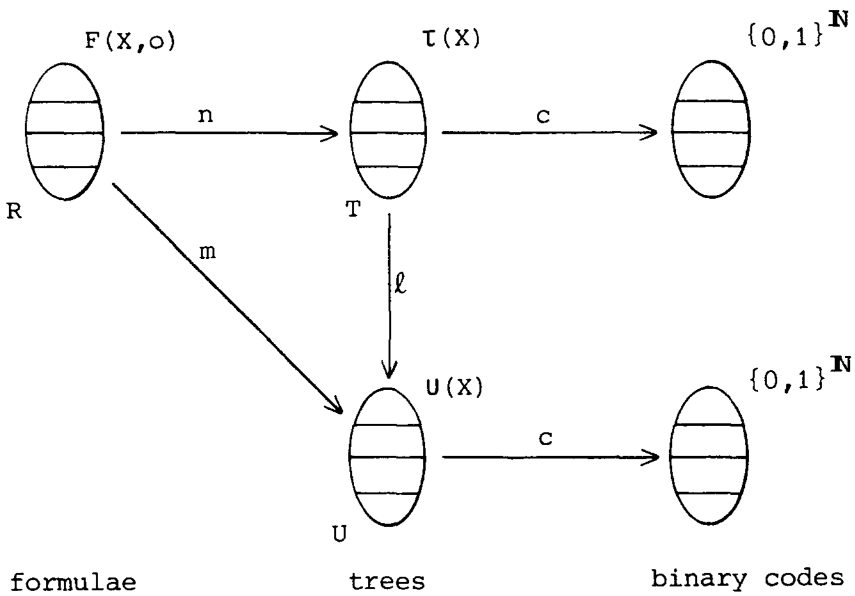

Fig. 1

regard the corresponding equivalence relation, which is of the above-mentioned type if one reads the formulae from right to left.

In the following sections we shall describe mappings connected with such a relation R on $F(X, \circ)$. In particular, we shall find a mapping from the quotient $F(X, \circ)/R$ to a set of labeled, oriented trees. Since the mappings to be described are bijections, they can be of aid in enumeration problems concerning formulae and trees. The main results of this paper can be found in Section 4. The final Section 5 contains a small excursion to the *binary coding* of trees.

As a mnemonic aid, we condense the sets, bijections and relations to be described in a diagram (Fig. 1).

2. Definitions and operations concerning trees

We consider ordered (hence rooted) trees which are *partially labeled* with elements of X, i.e., each node in a tree may be labeled with some symbol x from X, or may not be labeled. The set of all these trees will be called $\lambda(X)$. We define the following subsets of $\lambda(X)$. First, $\tau(X)$ is the set of all *trivalent* ordered trees of which all (and only the) *end* nodes have been labeled with elements of X. Secondly, $\upsilon(X)$ is the set of all ordered trees, not necessarily trivalent, of which *all* nodes have been labeled with elements of X. Finally, $\rho(X)$ is the set of all trees consisting solely of a labeled root. It is evident that $\rho(X) \subset \tau(X)$, $\rho(X) \subset \upsilon(X)$ and $\rho(X) = \tau(X) \cap \upsilon(X)$. For trees in $\lambda(X)$ we define the following operations. Let a be a tree in $\lambda(X)$. Then tree $\underset{\smile}{a}$ is obtained by *planting* tree a on a stem, i.e., by erasing the root-mark off the root-node of a and connecting this node by means of a single edge to a new, unlabeled root-node. Next, let a and b be trees in $\lambda(X)$ of which at most one root-node has been labeled. Then $[ab]$ is the ordered tree obtained by juxtaposing a and b in this order and identifying the root-nodes of a and b; if the root of either a or b was labeled, then this label is attached to the new, 'fused' root-node.

Example. In Fig. 2 we picture $\underline{a}$ and $[ab]$ for given a and b.

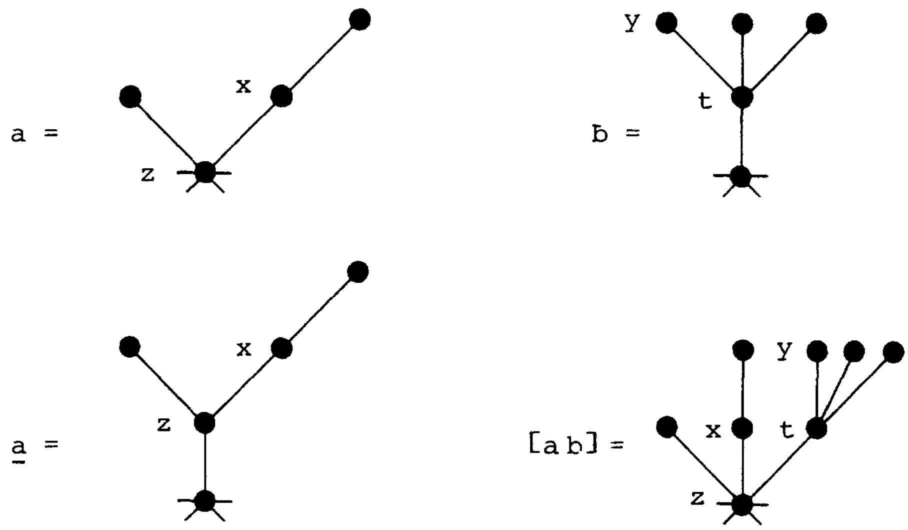

Fig. 2

Furthermore, for $x \in X$ we define $[x]$ to be the tree in $\rho(X)$ consisting solely of a root with label x. With these conventions we can give the following recursive *formula description* of $\tau(X)$:

(i) if $x \in X$, then $[x] \in \tau(X)$;

(ii) if a and $b \in \tau(X)$, then $[\underline{a}\,\underline{b}] \in \tau(X)$.

Finally, we define the equivalence relation U on $\lambda(X)$ by:
If a and $b \in \lambda(X)$, then aUb iff a and b are equal when considered as oriented (hence unordered) trees.

3. Coding of $F(X, \circ)$ by means of trees

We return to $F(X, \circ)$, the set of formulae on X and $\circ$. There is a well-known mapping n of $F(X, \circ)$ onto $\tau(X)$, recursively defined by:

(i) If $f = x \in X$, then $nf = [x]$;

(ii) If $f, g \in F(X, \circ)$, then $n(f \circ g) = [\underline{nf}\,\underline{ng}]$.

(Note that $[\underline{nf}\,\underline{ng}]$ is well-defined since $\underline{nf}$ has an unlabeled root.)

Examples. $n((x \circ y) \circ z) = [[[\underline{x}][\underline{y}]][\underline{z}]]$. (For a picture of the tree: see Fig. 3a.)

$n(x \circ (y \circ z)) = [[\underline{x}][[\underline{y}][\underline{z}]]]$. (See Fig. 3b.)

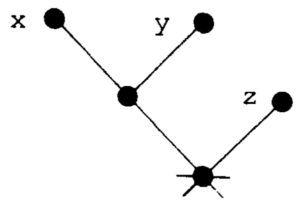

Fig. 3a

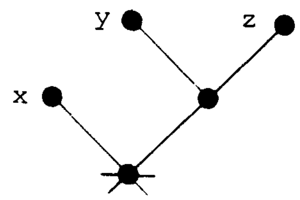

Fig. 3b

The mapping n is clearly a bijection (when we disregard the parameter $\circ$), which enables us to consider $nf \in \tau(X)$ as a coding of $f \in F(X, \circ)$. Since f and nf have a similar structure, we may regard n as a *natural* coding.

We shall also consider a second mapping of formulae, viz. a mapping of $F(X, \circ)$ onto $v(X)$. This mapping m is defined in a similar manner as the mapping n, but for one underlining in the definition:

(i) If $f = x \in X$, then $mf = [x]$;

(ii) If $f, g \in F(X, \circ)$, then $m(f \circ g) = [mf\,\underline{mg}]$.

(Note again that $[mf\,\underline{mg}]$ is well-defined.)

Examples. $m((x \circ y) \circ z) = [[[x][\underline{y}]][\underline{z}]]$. (For a picture of the tree: see Fig. 4a.)

$m(x \circ (y \circ z)) = [[x]\underline{[[y][\underline{z}]]}]$. (See Fig. 4b.)

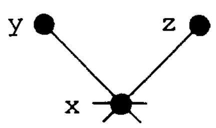

Fig. 4a

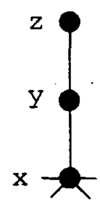

Fig. 4b

This mapping m is a bijection as well, which can be proved by induction. Hence, $mf \in v(X)$ is another *coding* of $f \in F(X, \circ)$.

Since both n and m are bijective mappings, there is a mapping l: $\tau(X) \to v(X)$ such that $l \circ n = m$ (viz. $l = m \circ n^{-1}$). We may obtain la directly from $a \in \tau(X)$ by erasing the left underlining in each pair of the form $[\underline{f}\,\underline{g}]$ occurring in the formula description of a; it holds that $l[\underline{f}\,\underline{g}] = [lf\,\underline{lg}]$.

As a mapping between sets of trees, the mapping l amounts to a process that we could call "a left branch contraction". We shall demonstrate this by means of an example, taking the trees coding the formula

$$f = (((x \circ y) \circ z) \circ (u \circ v)), \text{ where } x, y, z, u, v \in X.$$

In Fig. 5 we suggest how the tree $nf \in \tau(X)$ (at the left) is 'contracted' by mapping l into the tree $mf \in v(X)$ (at the right). Analogously, the mapping l^{-1}: $v(x) \to \tau(X)$ gives rise to a "left branch expansion".

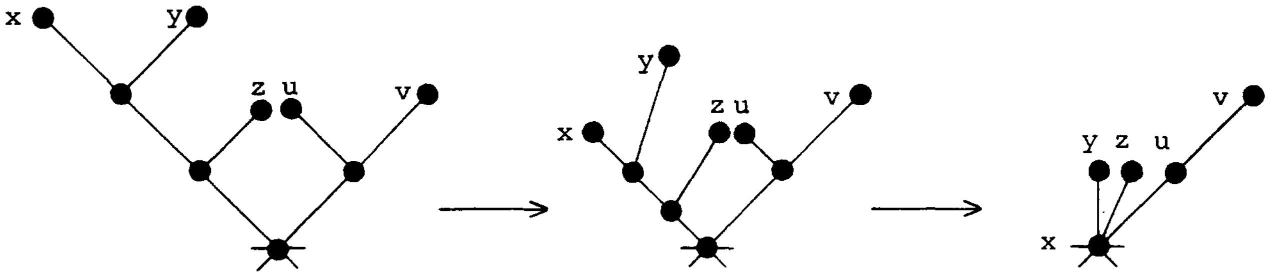

Fig. 5

The mapping l maps a tree in $\tau(X)$ with $2n-1$ vertices to a tree in $v(X)$ with n vertices. Hence, the mapping l yields a bijection, for each $n \in \mathbb{N}$, between the class of all trivalent ordered trees with $2n-1$ vertices and the class of all oriented trees with n vertices. The equipotency of these classes is well-known in the literature (see e.g. [2]). The mapping l described in this paper, however, is different from the usual bijections given in the literature for establishing this equipotency.

Note. The choice to define $m(f \circ g) = [mf\,\underline{mg}]$ is of course connected with relation R on $F(X, \circ)$, as will turn out in the sequel. When the relation R is replaced by the 'mirror-relation' R', obeying $(h \circ (g \circ f))R'(g \circ (h \circ f))$, then the mapping m' with $m'(f \circ g) = [\underline{mf}\,mg]$ would be appropriate. In that case, instead of mapping l, we have a mapping r: $\tau(X) \to v(X)$, which can be described as a "right branch contraction". The mappings l and r are, in a sense, mirror-images and of course have analogous properties.

4. Induced equivalence relations

It will be clear that the equivalence relation R on $F(X, \circ)$, defined in Section 1, induces, by means of mapping n, an equivalence relation T in $\tau(X)$. By the naturalness of n, relations R and T have much in common, particularly when the formula descriptions of the trees in $\tau(X)$ are considered; cf. $((x \circ y) \circ z)R((x \circ z) \circ y)$ and $[[\underline{[\underline{x}][\underline{y}]}][\underline{z}]]T[[\underline{[\underline{x}][\underline{z}]}][\underline{y}]]$.

It is less obvious which equivalence relation is induced by R by means of the mapping m. It will turn out that, this time, the induced relation is precisely U, the relation which 'identifies' ordered trees which differ only with respect to order (see Section 2). This is a consequence of the following theorem.

Theorem 1. *Let a and b be elements of $\tau(X)$. Then aTb iff $laUlb$.*

Proof. The only-if-part is proved by induction on the length of the proof of aTb. The main case is: $a = [[\underline{\underline{c}\,\underline{d}}]\,\underline{e}]$ and $b = [[\underline{\underline{c}\,\underline{e}}]\,\underline{d}]$. Then $la = [[c\underline{d}]\,\underline{e}]$ and $lb = [[c\underline{e}]\,\underline{d}]$, which trees only differ in order.

The if-part is proved by induction on $\max(\varepsilon a, \varepsilon b)$, where εa is the number of end-nodes in a. The main case is here: $a = [c\underline{d}]$ and $b = [e\underline{f}]$, so $la = [lc\, \underline{ld}]$ and $lb = [le\, \underline{lf}]$. Since $laUlb$, either

(i) $lcUle$ and $ldUlf$, or

(ii) $[lc\, \underline{ld}]U[le\, \underline{lf}]U[lg\, \underline{ld}\, \underline{lf}]$ for some $g \in \tau(X)$.

In the first case, by induction: cTe and dTf, hence aTb.

In the second case: $lcU[lg\, \underline{lf}]$ and $leU[lg\, \underline{ld}]$. By induction: $cT[g\underline{f}]$ and $eT[g\underline{d}]$, so $[c\underline{d}]T[[\underline{g\underline{f}}]\, \underline{d}]T[[\underline{g\underline{d}}]\, \underline{f}]T[e\underline{f}]$, hence aTb. □

Corollary 1. *There exists a bijection k from $\tau(X)/T$ to the set of oriented trees.*

Proof. If $C \in \tau(X)/T$, then take $k(C)$ to be the oriented tree corresponding to la for any $a \in C$. □

Corollary 2. *The classes of R-equivalent formulae in $F(X, \circ)$ can be coded by means of oriented trees.*

Proof. Let $G \in F(X, \circ)/R$. As a code of G we may take the oriented tree corresponding to mf for any $f \in G$. □

Note. The assertion stated in the latter corollary can be applied immediately to an enumeration problem considered in [1], thus simplifying the solution given there.

5. Corresponding transformations on the binary codings of trees

It is well known (see e.g. [3, p. 157]) that ordered trees can be coded by means of binary strings of zeros and ones. (We shall disregard the labels from X and the symbol $\circ$ in this section.) This coding can be defined recursively as follows: if an ordered tree consists of a root which bears n subtrees coded $C_1, \ldots, C_n$ (in order), then the tree as a whole is coded $0\ C_1 \cdots C_n\ 1$. A sole root is coded by 01.

There is a close and simple correspondence between the formula description of a tree, mentioned in Section 2, and the binary code presently mentioned: all we have to do is to read brackets '[' as zeros and brackets ']' as ones.

Example. The tree with formula description:

$$[[\underline{\underline{[[\underline{x}][\underline{y}]]}[\underline{z}]}][\underline{[\underline{u}][\underline{v}]}]],$$

which has been depicted in Section 3, has the binary code:

0 0 0 0 1 0 1 1 0 1 1 0 0 1 0 1 1 1.

In this section we shall describe what the effect is of mappings l and l^{-1} for the binary codes corresponding to trees in $\tau(X)$ and $v(X)$, respectively. (We shall not enter into the proofs of the procedures to be given.)

Let a be a tree in $\tau(X)$, with binary code ca. We can find the code cla from ca by means of the following procedure.

(i) Partition ca into parts of the form $0^k\ 1^l$, with $k \geqslant 1$ and $l \geqslant 1$. (Here 0^k denotes the concatenation of k zeros; etc.)

(ii) Replace each part $0^k\ 1^l$ thus obtained by $0\ 1^{l-1}$. (Here 1^0 stands for the empty string.)

(iii) Add a 1 at the end.

Example. Consider the tree mentioned above, with

$$ca = 0\ 0\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 1\ 1.$$

Then step (i) leads to: $0^4\ 1 \mid 0\ 1^2 \mid 0\ 1^2 \mid 0^2\ 1 \mid 0\ 1^3$, step (ii) yields $0 \mid 0\ 1 \mid 0\ 1 \mid 0 \mid 0\ 1^2$ and step (iii): $0\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 1$, which codes cla.

The inverse procedure, for constructing ca out of cla, is a little more complicated:

First, form the 0 1-nests in ca by connecting corresponding zeros and ones. Now insert a symbol 1 between each adjacent pair 1 0 in cla and insert a symbol string $0^m 1$ between each adjacent pair 0 0 in cla, m being the number of primary subnests of the nest corresponding to the first 0 of the pair 0 0 under consideration. The resulting code is ca.

Example. Take $cla = 0\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 1$, then the following nests can be found:

$$0\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 1.$$

Insertion of ones between adjacent pairs 1 0 leads to:

$$0\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 1$$

and the described insertion of strings of the form $0^m\ 1$ yields:

$$0\ 0^3\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 1\ 1.$$

Here, indeed, we have obtained ca.

References

[1] F. Göbel and R.P. Nederpelt, The number of numerical outcomes of iterated powers, Amer. Math. Monthly 78 (1971) 1097–1103.

[2] D.A. Klarner, Correspondences between plane trees and binary sequences, J. Combin. Theory 9 (1970) 401–411.

[3] R.C. Read, The coding of various kinds of unlabeled trees, in: R.C. Read, ed., Graph Theory and Computing (Academic Press, New York, 1972).